

## CLAIMS

1. A method for distributed computation of an RSA inverse value ( $y$ ) in an asynchronous network from at least two input values ( $x, e$ ) among  $n-1$  participating network devices comprising  $t < n/4$  faulty devices and a non-faulty leader device, the participating network  
5 devices holding share values of the Euler function ( $\phi(N)$ ) of an RSA modulus ( $N$ ), each participating network device performing the steps of:
  - (a) choosing a first random value ( $q$ ) and a second random value ( $r$ );
  - (b) sharing over integers ( $\mathcal{Z}$ ) the first random value ( $q$ ), the second random value ( $r$ ), and the zero value ( $0$ );
  - 10 (c) the leader device performing additionally the steps of:
    - (i) receiving a first, second, and third sub-share value ( $q_i, r_i, 0_i$ ) from at least  $t + 1$  participating network devices;
    - (ii) broadcasting the identities of said participating network devices;
  - (d) receiving the identities and corresponding sub-share values ( $q_i, r_i, 0_i$ );
  - 15 (e) deriving a sum-share value ( $F$ ) from one of the share values, the at least one input value ( $e$ ), and the corresponding first, second and third sub-share values ( $q_i, r_i, 0_i$ ) defined by the identities,
  - (f) broadcasting the sum-share value ( $F$ );
  - (g) receiving  $2t + 1$  sum-share values ( $F_i$ );
  - 20 (h) deriving a polynomial ( $f$ ) interpolating the sum-share values ( $F_i$ ) and an exponent share value ( $d_p$ ) dependent on the polynomial ( $f$ ), and an inverse-share value ( $y_p$ ) dependent on the exponent share value ( $d_p$ ) and the RSA modulus ( $N$ );
  - (i) broadcasting the inverse-share value ( $y_p$ );
  - (j) receiving  $t + 1$  inverse-share values ( $y_i$ ); and
  - 25 (k) obtaining the RSA inverse value ( $y$ ) from the received inverse-share value ( $y_i$ ).

2. A method according to claim 1 to compute the RSA inverse value ( $y$ ), wherein among  $n$  participating network devices (A, B, C, D) comprising  $t < n/4$  faulty devices at least  $t + 1$  participating network devices (A, B, C, D) act as a leader device while performing  $n$  times the steps of claim 1.
- 5 3. A method according to claim 2, wherein each participating network device (A, B, C, D) performs the steps of:
  - in the event of obtaining or receiving one RSA inverse value ( $y$ ), determining the validity of the obtained RSA inverse value ( $y$ ) under use of the at least two input values ( $x$ ,  $e$ ), and
  - 10 in the event of positive determination, broadcasting the RSA inverse value ( $y$ ) and stopping further calculations.
4. A method according to claim 1, wherein the step (II) of sharing over integers ( $Z$ ) comprises using a threshold signature for determining the consistency of subsequently received sub-share values ( $q_i, r_i, 0_i$ ).
- 15 5. A method according to claim 1, wherein the step (II) of sharing over integers ( $Z$ ) comprises using a vector of digital signatures for determining the consistency of subsequently received sub-share values ( $q_i, r_i, 0_i$ ).
6. A method according to claim 1, wherein the step (VIII) of deriving an exponent share value ( $d_p$ ) comprises using the Extended Euclidean Algorithm.
- 20 7. A method according to claim 1, wherein the step (XI) of obtaining the RSA inverse value ( $y$ ) from the received inverse-share value ( $y_i$ ) comprises using the Lagrange Interpolation Algorithm.
8. A plurality of program storage devices, each readable by at least one digital processing apparatus and having a program of instructions which are tangibly embodied in the

storage devices and which are executable by the at least one processing apparatus to perform a method for distributed computation of an RSA inverse value ( $y$ ), in an asynchronous network from at least two input values ( $x, e$ ) among  $n-1$  participating network devices (A, B, C) comprising  $t < n/4$  faulty devices and a non-faulty leader device (D), the participating network devices (A, B, C, D) holding share values ( $\varphi_A, \varphi_B, \varphi_C, \varphi_D$ ) of the Euler function ( $\varphi(N)$ ) of an RSA modulus ( $N$ ), said program allowing each participating network device to perform the following steps:

(a) choosing a first random value ( $q$ ) and a second random value ( $r$ );

(b) sharing over integers ( $\mathcal{Z}$ ) the first random value ( $q$ ), the second random value ( $r$ ), and the zero value (0);

(c) the leader device (D) performing additionally the steps of:

(i) receiving a first, second, and third sub-share value ( $q_i, r_i, 0_i$ ) from at least  $t + 1$  participating network devices;

(ii) broadcasting the identities ( $\mathcal{S}$ ) of said participating network devices;

(d) receiving the identities ( $\mathcal{S}$ ) and corresponding sub-share values ( $q_i, r_i, 0_i$ );

(e) deriving a sum-share value ( $F$ ) from the share value ( $\varphi_P$ ), the at least one input value ( $e$ ), and the corresponding sub-share values ( $q_i, r_i, 0_i$ ) defined by the identities ( $\mathcal{S}$ ),

(f) broadcasting the sum-share value ( $F$ );

(g) receiving  $2t + 1$  sum-share values ( $F_i$ );

(h) deriving a polynomial ( $f$ ) interpolating the sum-share values ( $F_i$ ) and an exponent share value ( $d_P$ ) dependent on the polynomial ( $f$ ), and an inverse-share value ( $y_P$ ) dependent on the exponent share value ( $d_P$ ) and the RSA modulus ( $N$ );

(i) broadcasting the inverse-share value ( $y_P$ );

(j) receiving  $t + 1$  inverse-share value ( $y_i$ ); and

(k) obtaining the RSA inverse value ( $y$ ) from the received inverse-share value ( $y_i$ ).